Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method of operating an n-dimensional array of processing elements to determine a dimensional extrema for a plurality of values stored in said n-dimensional array of processing elements, the method comprising:

determining within each of said processing elements a local extrema for each of said processing elements, said local extrema having a most significant byte and a least significant byte;

serially outputting in bursts said most significant bytes and said least significant bytes of said local extrema from each of said processing elements to a neighboring processing element until every processing element in a first dimension has received all local extrema along said first dimension, wherein a burst length is selected to optimize use of each processing element's ALU;

determining within each of said processing elements a first dimensional extrema for said first dimension of said n-dimensional array, wherein said first dimensional extrema is determined from a plurality of local extrema most significant bytes and least significant bytes stored in-said processing elements in said first dimension and wherein said first dimensional extrema has a most significant byte and a least significant byte; and

saving in a register said dimensional extrema.

2. (currently amended) The method of claim ~~11~~ 1 wherein said determining within each of said processing elements a first dimensional extrema for a first dimension of said n-dimensional array comprises:

receiving a set of local extrema from said processing elements within said first dimension;

separating said set of local extrema into an odd set corresponding to values in odd positions within said set of received local extremas and an even set corresponding to values in even positions within said set of received local extremas;

separating each of said odd local extrema into at least one of an odd most significant byte and an odd least significant byte;

separating each of said even local extrema into at least one of an even most significant byte and an even least significant byte;

determining an odd extrema from said odd set of least significant bytes and most significant bytes;

determining an even extrema from said even set of least significant bytes and most significant bytes; and

determining said first dimensional extrema for a first dimension from said odd extrema and said even extrema.

3. (previously presented) The method of claim 2 wherein said receiving a set of local extrema from said processing elements within said first dimension comprises:

receiving a burst of said odd and even least significant bytes; and

receiving a burst of said odd and even most significant bytes.

4. (original) The method of claim 3 further comprising:

selecting a burst length for said burst of odd and even least significant bytes and a burst length for said odd and even most significant bytes to minimize the amount of lost cycles within said processing elements.

5. (previously presented) The method of claim 1 additionally comprising determining within each of said processing elements a next dimensional extrema for a next dimension of said n-dimensional array, said determining a next dimensional extrema comprising:

receiving a set of said first dimensional extrema from said processing elements within said next dimension;

separating said set of first dimensional extrema into an odd set corresponding to values in odd positions within said set of received first dimensional extrema and an even set corresponding to values in even positions within said set of received first dimensional extrema;

separating each of said odd first dimensional extrema into at least one of an odd most significant byte and an odd least significant byte;

separating each of said even first dimensional extrema into at least one of an even most significant byte and an even least significant byte;

determining an odd extrema from said odd set;

determining an even extrema from said even set; and

determining said next dimensional extrema for a next dimension from said odd extrema and said even extrema.

6. (previously presented) The method of claim 5 wherein said receiving a set of said first dimensional extrema from said processing elements within said next dimension comprises:

receiving a burst of said odd and even least significant bytes; and

receiving a burst of said odd and even most significant bytes.

7. (original) The method of claim 6 further comprising:

selecting a burst length for said burst of odd and even least significant bytes and a burst length for said odd and even most significant bytes to minimize the amount of lost cycles within said processing elements.

8. (previously presented) The method of claim 1 additionally comprising repeating said determining within each of said processing elements a next dimensional extrema for each of n-said dimensions, said repeating said determining comprising:

receiving a set of dimensional extrema from a previously selected dimension from said processing elements within a currently selected dimension;

separating said set of dimensional extrema from said previously selected dimension into an odd set corresponding to values in odd positions within said set of received next dimensional extrema and an even set corresponding to values in even positions within said set of received next dimensional extrema;

separating each of said odd next dimensional extrema into at least one of an odd most significant byte and an odd least significant byte;

separating each of said even next dimensional extrema into at least one of an even most significant byte and an even least significant byte;

determining a odd extrema from said odd set;

determining an even extrema from said even set; and

determining said next dimensional extrema for said next dimension from said odd extrema and said even extrema.

9. (previously presented) The method of claim 2 wherein determining an odd extrema from said odd set comprises:

loading the least significant byte of an odd local extrema into a least significant odd byte register;

loading the most significant byte of said odd local extrema into a most significant odd byte register;

comparing the contents of said least significant odd byte register to the least significant byte of another odd local extrema and setting a carry flag relative to said comparison;

comparing the contents of said most significant odd byte register to the most significant byte of said another odd local extrema and setting an odd flag relative to said comparison; and

conditionally updating said most significant odd byte register and said least significant odd byte register relative to said odd flag.

10. (previously presented) The method of claim 2 wherein said determining an even extrema from said even set comprises.

loading the least significant byte of an even local extrema into a least significant even byte register;

loading the most significant byte of said even local extrema into a most significant even byte register;

comparing the contents of said least significant even byte register to the least significant byte of another even local extrema and setting a carry flag relative to said comparison;

comparing the contents of said most significant even byte register to the most significant byte of said another even local extrema and setting an even flag relative to said comparison; and

conditionally updating said most significant even byte register and said least significant even byte register relative to said even flag.

11. (previously presented) The method of claim 9 further comprising repeating said comparing the contents of said most significant odd byte register, said comparing the contents of said least significant odd byte register, and said conditionally updating said most significant odd byte register and said least significant odd byte register for each of said odd local extrema within said set.

12. (previously presented) The method of claim 10 further comprising repeating said comparing the contents of said most significant even byte register, said comparing the contents of said least significant even byte register, and said conditionally updating said most significant even byte register and said least significant even byte register for each of said even local extrema within said set.

13. (previously presented) The method of claim 2 wherein said determining said first dimensional extrema from said odd extrema and said even extrema further comprises:

loading the least significant byte of said odd extrema into a least significant odd byte register and the most significant byte of said odd extrema into a most significant odd byte register;

loading the least significant byte of said even extrema into a least significant even byte register and the most significant byte of said even extrema into a most significant even byte register;

comparing the contents of said least significant even byte register to the contents of said least significant odd byte register and setting a carry flag relative to a result of said comparison;

comparing the contents of said most significant even byte register to the contents of said most significant odd byte register and setting an extrema flag relative to a result of said comparison; and

conditionally updating said most significant even byte register and said least significant even byte register relative to said extrema flag.

14.–30. Cancelled.

31. (previously presented) An n-dimensional array of processing elements, comprising:

a plurality of processing elements interconnected to form an n-dimensional array, each processing element comprising:

an arithmetic logic unit;

condition logic responsive to said arithmetic logic unit;

a plurality of registers connected to a bus and responsive to said arithmetic logic unit;

a result pipeline responsive to said arithmetic logic unit;

an interface; and

register files connected between said interface and said result pipeline; said processing elements configured to:

determine a local extrema for each of said processing elements, said local extrema having a most significant byte and a least significant byte;

serially output in bursts said most significant bytes and said least significant bytes of said local extrema from each of said processing elements to a neighboring processing element until every processing element in a first dimension has received all local extrema along said first dimension, where a burst length is selected to optimize use of each processing elements arithmetic logic unit;

determine within each of said processing elements a first dimensional extrema for said first dimension of said n-dimensional array, wherein said first dimensional extrema is determined from a plurality of local extrema most significant bytes and least significant bytes stored in said processing elements in said first dimension and wherein said first dimensional extrema has a most significant byte and a least significant byte; and

saving said dimensional extrema.

32. (previously presented) The array of processing elements of claim 31 wherein said processing elements are configured to:

receive a set of local extrema from said processing elements within said first dimension;

separate said set of local extrema into an odd set corresponding to values in odd positions within said set of received local extremas and an even set corresponding to values in even positions within said set of received local extremas;

separate ach of said odd local extrema into at least one of an odd most significant byte and an odd least significant byte;

separate each of said even local extrema into at least one of an even most significant byte and an even least significant byte;

determine an odd extrema from said odd set of least significant bytes and most significant bytes;

determine an even extrema from said even set of least significant bytes and most significant bytes; and

determine said first dimensional extrema for a first dimension from said odd extrema and said even extrema.

33. (previously presented) The array of processing elements of claim 32 wherein said processing elements are configured to:

receive a burst of said odd and even least significant bytes; and

receive a burst of said odd and even most significant bytes,

and wherein a burst length for said burst of odd and even least significant bytes and a burst length for said odd and even most significant bytes is selected to minimize the amount of lost cycles within said processing elements.

34. (previously presented) The array of processing elements of claim 31 wherein said processing elements are additionally configured to:

receive a set of said first dimensional extrema from said processing elements within a next dimension;

separate said set of first dimensional extrema into an odd set corresponding to values in odd positions within said set of received and an even set corresponding to values in even positions within said set of received;

separate each of said odd first dimensional extrema into at least one of an odd most significant byte and an odd least significant byte;

separate each of said even first dimensional extrema into at least one of an even most significant byte and an even least significant byte;

determine an odd extrema from said odd set;

determine an even extrema from said even set; and

determine said next dimensional extrema for a next dimension from said odd extrema and said even extrema.

35. (previously presented) The array of processing elements of claim 34 wherein said processing elements are configured to:

receive a burst of said odd and even least significant bytes; and

receive a burst of said odd and even most significant bytes,

and wherein a burst length for said burst of odd and even least significant bytes and a burst length for said odd and even most significant bytes is selected to minimize the amount of lost cycles within said processing elements.

36. (previously presented) The array of processing elements of claim 31 wherein said processing elements are additionally configured to:

receive a set of next dimensional extrema from a previously selected dimension from said processing elements within a currently selected dimension;

separate said set of dimensional extrema from said previously selected dimension into an odd set corresponding to values in odd positions within said set of received next dimensional extrema and an even set corresponding to values in even positions within said set of received next dimensional extrema;

separate each of said odd next dimensional extrema into at least one of an odd most significant byte and an odd least significant byte;

separate each of said even next dimensional extrema into at least one of an even most significant byte and an even least significant byte;

determine a odd extrema from said odd set;

determine an even extrema from said even set; and

determine said next dimensional extrema for said next dimension from said odd extrema and said even extrema.

37. (previously presented) The array of processing elements of claim 32 wherein said processing elements are configured to:

load the least significant byte of an odd local extrema into a least significant odd byte register;

load the most significant byte of said odd local extrema into a most significant odd byte register;

compare the contents of said least significant odd byte register to the least significant byte of another odd local extrema and setting a carry flag relative to said comparison;

compare the contents of said most significant odd byte register to the most significant byte of said another odd local extrema and setting an odd flag relative to said comparison; and

conditionally update said most significant odd byte register and said least significant odd byte register relative to said odd flag.

38. (previously presented) The array of processing elements of claim 32 wherein said processing elements are configured to:

load the least significant byte of an even local extrema into a least significant even byte register;

load the most significant byte of said even local extrema into a most significant even byte register;

compare the contents of said least significant even byte register to the least significant byte of another even local extrema and setting a carry flag relative to said comparison;

compare the contents of said most significant even byte register to the most significant byte of said another even local extrema and setting an even flag relative to said comparison; and

conditionally update said most significant even byte register and said least significant even byte register relative to said even flag.

39. (previously presented) The array of processing elements of claim 37 wherein said processing elements are configured to:

repeat said comparing the contents of said most significant odd byte register, said comparing the contents of said least significant odd byte register, and said conditionally updating said most significant odd byte register and said least significant odd byte register for each of said odd local extrema within said set.

40. (previously presented) The array of processing elements of claim 38 wherein said processing elements are configured to:

repeat said comparing the contents of said most significant even byte register, said comparing the contents of said least significant even byte register, and said conditionally updating said most significant even byte register and said least significant even byte register for each of said even local extrema within said set.

41. (previously presented) The array of processing elements of claim 32 wherein said processing elements are configured to:

load the least significant byte of said odd extrema into a least significant odd byte register and the most significant byte of said odd extrema into a most significant odd byte register;

load the least significant byte of said even extrema into a least significant even byte register and the most significant byte of said even extrema into a most significant even byte register;

compare the contents of said least significant even byte register to the contents of said least significant odd byte register and setting a carry flag relative to a result of said comparison;

compare the contents of said most significant even byte register to the contents of said most significant odd byte register and setting an extrema flag relative to a result of said comparison; and

conditionally update said most significant even byte register and said least significant even byte register relative to said extrema flag.